Learning points

This project has made me familiar with the various parts of a web app. Rails was a little overwhelming to me at first, with so many moving parts—from MVC, assets, databases, and tests, to deployment—but it eventually made sense as I came to understand (through practice) what the various components do, and how they interact with one another in a meaningful way.

Here are some things that I've come to understand better along the way:

The MVC model. I've tried working with frameworks in other languages before, but the projects were so tiny that all the code could be squeezed into one or two files. Learning the MVC model here has helped to enforce modularity (which can be useful when dealing with even more complexity), and the idea of separation of concerns can also be applied elsewhere, across different languages and frameworks.

Git. I've used Git before, but I've now gained more familiarity with it and have learned to use some new features (such as `git commit --amend`, `git stash`, and `git rebase -i`) as a result of making mistakes with my commits along the way.

Relational databases. Rails already provides a neat abstraction to deal with the complexity of interacting with databases, but I've learned about them to a limited extent nonetheless. I didn't get to try modelling relations in this app though, and would like to try them sometime still.

Designing UIs. I've tried to make interacting with the app as intuitive as possible, in line with the principle of least astonishment. Some choices include using icons that make their functions obvious (star, pencil, trash etc.), and designing everything on a single page with the help of the front-end framework Bootstrap (which has made design much more tolerable).

Unit tests. I wrote a few tests at the end, mostly for the sake of learning how to write them, which was regrettable because they're generally more useful during development and not after (if I'm not wrong!). Writing tests as I went along would've given me more confidence to change things around, instead of wasting time manually testing every change I made to the codebase.

Keeping a project to-do list. It can be very helpful to keep track of new features to add, and issues to fix.

Other notes

My use of Ajax amounted to simply removing `local: true` on forms and adding `remote: true` on buttons, which was a bit of a let down because I thought I'd get to try writing some more JavaScript/jQuery here.

The app could also have been more ambitious, emulating other todos like Evernote or Org mode. It would've been interesting as I'd never tried anything too big before, and also get to implement features like deadlines and sorting criteria for todos along the way, but it might be overkill for an assignment.

User manual

This app is a todo manager, where you may add, edit, and delete todo items.

To *add* a new todo item, use the form in (1). The item will then be displayed in the list below.

To *edit* a todo item, hover over the item and click on its pencil icon (2). A modal will then appear with a form to edit the item.

To *delete* a todo item, click on its trash icon (3).

Todo items may also come with descriptions, which can be displayed by clicking on the item. To *edit* the description, use the same pencil icon in (2).

Actions

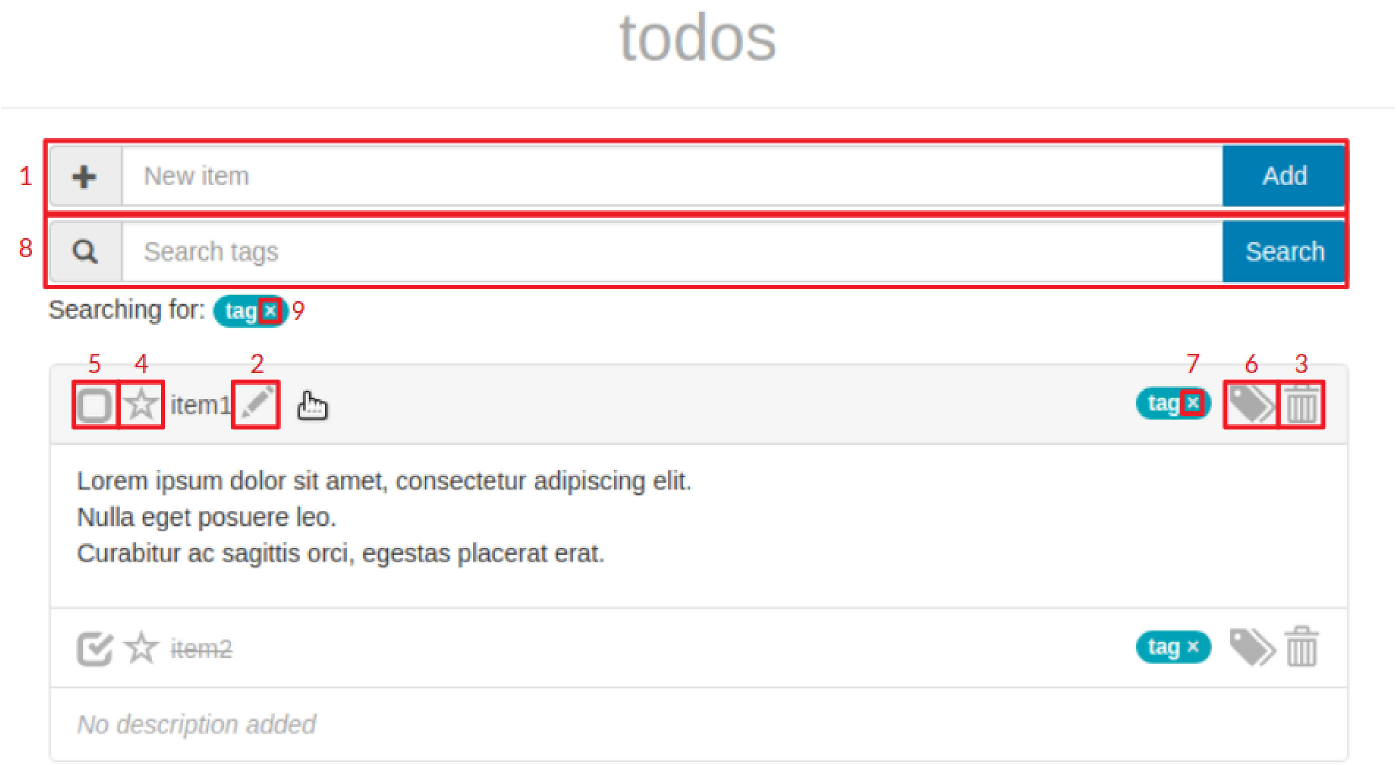
Todo items can also be prioritized or completed.

To *give priority* to a todo item, click on its star icon (4). The star will turn yellow, and the item will rise to the top, along with other prioritized items.

To *remove priority*, simply click on the star again.

To *complete* a todo item, click on its checkbox (5). The text will be striked through, and the item will be pushed to the bottom. Completed items may not be given priority.

To re-mark it as *incomplete*, simply click on the checkbox again.



Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nulla eget posuere leo.
Curabitur ac sagittis orci, egestas placerat erat.

No description added

Tags

Each todo item may be given multiple tags.

To *add* a tag to a todo item, click on its tag icon (6). A modal will then appear with a form to add a tag. The tag will then be displayed at the side of the item.

To *remove* a tag from a todo item, click on the cross beside it (7).

To *search* for todo items with a tag, use the form in (8). The list below will then be filtered to display only items with the tag.

To *remove* a search tag, click on the cross beside it (9).